

GCC224 - Introdução aos Algoritmos

Lista de Exercícios 9

Ponteiros e Alocação Dinâmica de Memória

1. (BACKES, 2012) Escreva um programa que contenha duas variáveis inteiras. Leia essas variáveis do teclado. Em seguida, compare seus endereços e exiba o conteúdo do maior endereço.
2. Faça um programa que declare duas variáveis double e solicite os valores das mesmas ao usuário. Em seguida, guarde o valor da soma em memória alocada dinamicamente e exiba o valor a partir de lá.
3. Faça um programa que declare uma variável inteira tamanho e um vetor v com esse tamanho. Em seguida guarde em variáveis:
 - a. o endereço da variável tamanho.
 - b. o endereço do vetor v.
 - c. o endereço do primeiro elemento de v.
 - d. o valor do primeiro elemento de v.
 - e. o endereço do último elemento de v.

Por fim, escreva o valor de cada uma dessas variáveis.

4. Quais atribuições abaixo são válidas e por que (ou por que não):

- a) `int numero;`
`int ptrNumero = №`
- b) `int ptrNumero = new int;`
- c) `int x, *ptr = &x;`
- d) `int v[5] = { 0, 1, 2, 3, 4 };`
`int *ptr;`
`*ptr = v[1];`
- e) `int v[5] = { 0, 1, 2, 3, 4 };`
`int *ptr;`
`ptr = &v[1];`
- g) `int v[5] = { 0, 1, 2, 3, 4 };`
`int a;`
`int *ptr = &a;`
`ptr = v[1];`
- f) `int v[5] = { 0, 1, 2, 3, 4 };`
`int *ptr = new int;`
`*ptr = v[1];`

- h) `int v[5] = { 0, 1, 2, 3, 4 };
int a;
int *ptr = &a;
*ptr = v[1];`
5. Escreva um programa que leia duas cadeias de caracteres e concatene a segunda cadeia ao final da primeira. As duas cadeias de entrada devem ser declaradas como vetores de char com tamanhos máximos fixos. A cadeia resultante deve ser declarada como um ponteiro para o tipo de dado char que aponta para uma variável dinâmica.
Obs: a função `strlen` retorna o número de caracteres úteis de um vetor de char.
6. Para cada opção abaixo o que seria exibido na tela se ao final fosse acrescentado o comando: `cout << r << endl;`
- a) `int x = 5, y = 2;
int *z = &x;
int *w = new int;
*w = *z;
int r = x + y + *w + *z;`
- b) `bool b1 = true, b2 = false, b3 = true;
bool *c = &b2;
bool d = b3;
bool r = b1 and *c and d;`
- c) `float a = 2.3, *b = &a, c = 4.5;
float v[3] = { 1.1, 2.2, 3.3}
float *x = new float[3];
float *y = v;
x[0] = a;
x[1] = *b;
x[2] = y[2] + c;
float r = x[0] + x[1] + x[2];`
7. Elabore um subprograma que calcule a soma dos elementos de uma matriz LxC de números reais (double). A implementação do programa deve considerar que as dimensões da matriz serão fornecidas pelo usuário. Represente a matriz por meio de uma variável `double**`, usando alocação dinâmica de memória.
8. Um ponteiro pode ser usado para dizer a um subprograma onde ele deve armazenar o resultado de seus cálculos. Escreva um subprograma **horasMinutos** que converta minutos em horas-e-minutos. O subprograma recebe um inteiro **mnts** e os endereços de duas variáveis inteiras, por exemplo **h** e **m**, e atribui valores a essas variáveis de modo que **m** seja menor que 60 e que $60 \cdot h + m$ seja igual a **mnts**. Faça um subprograma principal para testar o subprograma **horasMinutos**.
9. (BACKES, 2012) Crie um registro chamado Cadastro. Esse registro deve conter o nome, a idade e o endereço de uma pessoa. Agora, escreva uma função que receba um inteiro positivo N e retorne o ponteiro para um vetor de N Cadastros, alocado

dinamicamente. Solicite também que o usuário digite os dados desse vetor dentro da função.

10. Declare um registro para armazenar um resultado de um sorteio da mega sena. Faça um programa que receba do usuário uma quantidade de sorteios e as dezenas sorteadas, armazenando-as em um vetor alocado dinamicamente. Faça então um subprograma que receba esse vetor e retorne um ponteiro para o registro que tenha a maior soma de dezenas sorteadas. Por fim, exiba os dados do registro retornado.
11. (BACKES, 2012) Escreva um programa que aloque dinamicamente uma matriz de inteiros. As dimensões da matriz deverão ser lidas do usuário. Em seguida, escreva uma função que receba um valor e retorne 1, caso o valor esteja na matriz, ou retorne 0, no caso contrário.
12. (BACKES, 2012) Escreva uma função que receba como parâmetro uma matriz A 5×5 . A função deve retornar o ponteiro para um vetor B de tamanho 5 alocado dinamicamente, em que cada posição de B é a soma dos números da coluna correspondente da matriz.
13. Escreva uma função que recebe um número entre 1 e 12 e retorna o nome do mês correspondente. O retorno deve ser na função em si e não num dos parâmetros normais da função. A função não pode usar seletores. A função não pode usar o tipo string. Sugestões: a) Use um vetor de nomes (ponteiro para ponteiro para caractere). b) Faça alocação dinâmica na função para o resultado e deixe que a memória seja liberada fora da função.